What is claimed is:

1. A method of creating a ranked join index for ordered data entries, comprising:

determining a dominating set of said ordered data entries;

mapping said dominating set of ordered data entries according to rank attributes;

determining a separating vector for each set of adjacent mapped data entries; and

ordering and indexing said data entries according to a separating point associated with each of said separating vectors.

2. The method of claim 1, wherein determining the dominating set of said ordered data entries, comprises:

maintaining a priority queue of a predetermined size of said ordered data entries according to rank attributes, wherein data entries having the highest combined rank attribute values are maintained in said priority queue;

wherein, if said priority queue has reached a maximum capacity, only data entries having combined rank attribute values greater than the attribute value of data with a minimum rank value present in the queue are added to the queue.

3. The method of claim 2, wherein said predetermined size corresponds to a minimum number of data entries necessary to generate, in the worst case, a ranked join index providing answers with a desired guaranteed performance on any top-k join query.

4. The method of claim 1, wherein ordering and indexing said data entries, comprises:

sweeping a vector across a plane of said mapped data entries, wherein each time said vector crosses a separating vector, a current composition of highest ranked data entries changes by swapping at least one of the data entries in the adjacent data entries set if it causes a change in the index; and

wherein each time a data entry is swapped, the highest ranked data entries are materialized and a new index entry is initiated.

5. The method of claim 1, wherein said indexing is query independent such that substantially any user preference query may be resolved using said index.

6. The method of claim 1, wherein said adjacent data entry set comprises more than two mapped data points if data entries are collinear.

7. The method of claim 1, wherein performance guarantees are provided for an amount of time said index requires for resolving user queries.

8. The method of claim 1, further comprising merging ordered data entries.

9. The method of claim 8, wherein said merging results in a storage space requirement for said index that is characterized according to the following equation:

$$O(nK^2(K+m)/m)$$

wherein n represents a total number of data entries to be indexed, K represents an upper bound on a number of high ranking data entries that may be requested by a user, and m represents a total number of data entries to be merged.

10. The method of claim 8, wherein said merging results in a query time for said index that is characterized according to the following equation:

$$O(\log(nK^2/m)+(K+m)\log(K+m))$$

wherein n represents a total number of data entries to be indexed, K represents an upper bound on a number of high ranking data entries that may

be requested by a user, and m represents a total number of data entries to be merged.

11. The method of claim 8, wherein merging said ordered data entries provides space and time tradeoffs.

12. The method of claim 11, wherein a space and time tradeoff comprises reducing query time of said index by increasing storage space of said index.

13. The method of claim 11, wherein a space and time tradeoff comprises reducing storage space required by said index by increasing query time of said index.

14. A method of providing solutions to top-k join queries of ranked data entries for user specified preferences, comprising:

determining a dominating set of said ranked data entries;

mapping said dominating set of ranked data entries according to rank attributes;

creating a ranked join index for said ranked data entries by determining a separating vector for each set of adjacent mapped data entries;

ordering and indexing said data entries according to a separating point associated with each of said separating vectors; and

providing a solution for said user preference query using said ranked join index.

15. A computer-readable medium for storing a set of instructions, which when executed by a processor, perform a method of creating a ranked join index for ordered data entries, comprising:

determining a dominating set of said ordered data entries;

mapping said dominating set of ordered data entries according to rank attributes;

determining a separating vector for each set of adjacent mapped data entries; and

ordering and indexing said data entries according to a separating point associated with each of said separating vectors.

16. A computer-readable medium for storing a set of instructions, which when executed by a processor, perform a method of providing solutions to top-k join queries of ranked data entries for user specified preferences, comprising:

determining a dominating set of said ranked data entries;

mapping said dominating set of ranked data entries according to rank attributes;

creating a ranked join index for said ranked data entries by determining a separating vector for each set of adjacent mapped data entries;

ordering and indexing said data entries according to a separating point associated with each of said separating vectors; and

providing a solution for said user preference query using said ranked join index.

17. An apparatus, comprising a memory for storing information and program instructions and a processor for executing said instructions, said apparatus adapted to perform a method of creating a ranked join index for ordered data entries by performing the steps of:

determining a dominating set of said ordered data entries;

mapping said dominating set of ordered data entries according to rank attributes;

determining a separating vector for each set of adjacent mapped data entries; and

ordering and indexing said data entries according to a separating point associated with each of said separating vectors.

18. The apparatus of claim 17, wherein for determining the dominating set of said ordered data, said apparatus is adapted to perform the step of:

maintaining a priority queue of a predetermined size of said ordered data entries according to rank attributes, wherein data entries having the highest combined rank attribute values are maintained in said priority queue;

wherein, if said priority queue has reached a maximum capacity, only data entries having combined rank attribute values greater than the attribute value of data entries with a minimum rank value present in the queue are added to the queue.

19. The apparatus of claim 18, wherein the predetermined size of said priority queue corresponds to a minimum number of data entries necessary to generate, in the worst case, a ranked join index providing answers with a desired guaranteed performance on any top-k join query.